(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 942 370 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
15.09.1999 Bulletin 1999/37

(51) Int. Cl.6: G06F 9/46

(21) Application number: 99301899.3

(22) Date of filing: 12.03.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 13.03.1998 JP 06242098

(71) Applicant:
International Business Machines Corporation
Armonk, N.Y. 10504 (US)

(72) Inventors:
- Suzuki, Toshihiro
  Yokohama-shi, Kanagawa-ken (JP)
- Minami, Kazuhiro
  Yokohama-shi, Kanagawa-ken (JP)

(74) Representative:
Jennings, Michael John
IBM United Kingdom Limited,
Intellectual Property Department,
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(54) A mobile agent, a mobile agent object, and a method for allowing a mobile agent to identify characteristics of another mobile agent

(57) A mobile agent technology and method allow a mobile agent to identify its ancestor, children, grand children and sibling, etc., and to flexibly cooperate with them in working by implanting gene information in a mobile agent while maintaining a high security. The mobile agent is provided with ancestor gene information and self gene information. The self gene information is encrypted information which indicates how many generations the mobile agent is descendant from the original agent. When the mobile agent creates a clone (creates a child agent), it generates gene information of its self generation + 1 and implants it in the child agent. The mobile agent moving on the network and encountering another mobile agent on a same place obtains gene information of that agent. The mobile agent then convert the obtained gene information by a predetermined algorithm to identify that the agents were born from a same ancestor.

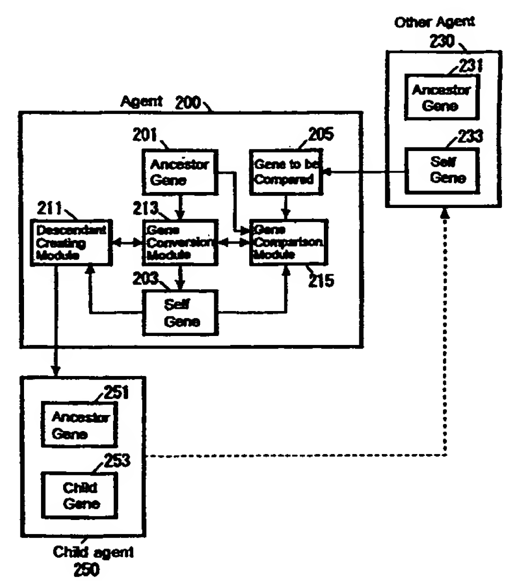


FIG. 3

Printed by Xerox (UK) Business Services
2.16.7/3.6

Description

Technical Field of the Invention

[0001] This invention relates to a mobile agent technology in which a mobile agent is moved to a remote server in a distributed computer network to generate an instruction on the remote server and to a method of moving a mobile agent to a remote server.

Background

[0002] There has been in the prior art a mobile agent technology to move a mobile agent to a remote server on a network to generate an instruction on the remote server in a distributed computer environment (PUPA 7-182174 (USP 5,603,031), PUPA 7-509799 (International Application Number PCT/US94/07397, International Publication Number:WO97/02219), "Latest Internet Technology", Nikkei Communication Separate Edition, Nikkei BP Co., pp 104 - 117, by Fumihiko Nishida, Susumu Fujiwara, et al).

[0003] Such mobile agent can be defined by two basic elements which are its mobile agent functions and the network location (hereafter "place") at which it performs its operations. The mobile agent can move around places existing on the network while maintaining its internal state. The mobile agent can contact another agent (a mobile agent or a non-mobile agent) in a place to receive a necessary service. The "place" is a place provided by a server existing on the network where the agent moves to, supports a contact between agents, and absorbs the complexities for interoperability which result from differences between hardware and operating system platforms.

[0004] Such mobile agent technology enables a mobile agent to execute jobs which have been done in the past by a human being, such as to adjust setting of an internal conference depending on a schedule of employees or a reservation status of a conference room and to obtain desired information distributed on the network.

[0005] There is a technology of classifying complex behavior of a mobile agent into basic behavior patterns, providing a template (moderator template) which enables a desired behavior pattern of the mobile agent to be defined, and controlling an activity of issuing a request in each place around which the mobile agent wanders as a predecessor (an activity working in a previous place) and a successor (an activity working in the next place), as disclosed in Japanese patent application 9-92091 filed by the applicant of this application on April 10, 1997 though not laid open at the time when this application is filed.

[0006] With this technology, a problem can be solved relying on the cooperativity of the mobile agent not only by moving the agent but also causing it to be split by itself and consolidating it. In this technology, however, the model is applied only to a predetermined cooperative algorithm (plan) and this technology has not come to a stage of providing a flexible cooperative problem solving technique which allows a cooperative algorithm defined beforehand to be reassembled or newly added.

[0007] This is because the mobile agent has no function to identify itself or a clone created from itself (children, grand children, sibling, cousin, uncle and aunt, etc), so that it can not determine and identify each other by an ID which is an index of split and consolidation of a mobile agent.

[0008] A simple self corroboration may be feasible by corroborating the identity by an ID in the form of a simple numerals but this may lead to a problem from a security view point in that data may be replaced with malicious data if the ID can be found by anybody.

Summary of the Invention

[0009] This invention provides a mechanism for allowing a mobile agent to identify its ancestor, children, grand children and sibling, etc., by implanting gene information in a mobile agent.

[0010] The invention preferably provides highly secured gene information which can not be understood by a third party.

[0011] The invention also preferably provides a mobile agent which can flexibly modify a job to be executed during movement.

[0012] The invention also preferably shortens a processing time required from the time when the mobile agent is released to the time when the result is returned.

[0013] A mobile agent is provided with ancestor gene information and self gene information. The self gene information is encrypted information indicating the number of generations separating the agent from the original agent. When a mobile agent creates a clone (a child agent), it creates gene information of its generation + 1 and implants it in the child agent. A mobile agent moving on the network and encountering another agent gets gene information of the latter agent and performs a given conversion to affirm that they are created from a same ancestor.

[0014] This invention provides, in one aspect thereof, a mobile agent which is adapted to interoperate with other mobile agents at the same network location, comprising:

(a) self gene information from which ancestor gene information and generation information can be derived by a predetermined conversion logic,

(b) a gene conversion module for generating child agent gene information from which said ancestor gene information and said generation information can be derived by a predetermined conversion logic, said child agent gene information being implanted in a child agent when creating a child

agent of said mobile agent, and

(c) a gene comparison module for determining whether or not the result of a conversion of gene information obtained from another agent corresponds to said ancestor gene information.

[0015] In the claims of this specification, the term "a predetermined conversion logic" is a concept including not only inverting a given bit string but also decoding encrypted information and decompressing compressed information, etc. The term "generation information" is a concept including not only information indicating how many generations a generation is descendant form a specific ancestor but also preferably information indicating whether a generation is more descendant than a certain generation, or information discriminating a descendant from other descendant with respect to a certain ancestor among ancestors (a head family or a branch family, a descendant of a specific sibling, etc.).

[0016] This invention provides, in another aspect thereof, an object held by a mobile agent which is adapted to interoperate with other mobile agents at the same network location, comprising:

(a) self gene information from which ancestor gene information and generation information can be derived by a predetermined conversion logic,

(b) a gene conversion module for generating child agent gene information from which said ancestor gene information and said generation information can be derived by a predetermined conversion logic, said child agent gene information being implanted in a child agent when creating a child agent of said mobile agent, and (c) a gene comparison module for determining whether or not the result of a conversion of gene information obtained from another agent corresponds to said ancestor gene information.

[0017] This invention provides, in further aspect thereof, a method of creating a child agent based on a mobile agent existing at a computer network location, the method comprising;

(a) a step of deriving generation information of said mobile agent based on self gene information and ancestor gene information,

(b) a step of generating gene information of a child agent based on said derived generation information and said ancestor gene information, and

(c) a step of generating a child agent which contains the generated gene information of the child agent and said ancestor gene information.

[0018] This invention provides, in a further aspect thereof, a method for allowing a first mobile agent existing at a computer network location to identify whether a second mobile agent existing at the same network location has the same ancestor gene information as is held in said first mobile agent, the method comprising;

(a) a step of obtaining gene information from said second mobile agent,

(b) a step of converting said obtained gene information by a predetermined logic, and

(c) a step of determining whether or not the result of the conversion corresponds to said ancestor gene information held in said first mobile agent.

Description of Preferred Embodiments

[0019] The present invention will now be described in more detail, by way of example, with reference to the accompanying drawings, in which:

Fig.1 shows a distributed network environment where a mobile object created in accordance with this invention operates.

Fig.2 shows a schematic diagram of a hardware configuration of node systems in a preferred embodiment of this invention.

Fig.3 is a functional block diagram showing an embodiment of processing components in the node system.

Fig.4 is an object diagram of the mobile agent created in a preferred embodiment of this invention.

Fig.5 is a diagram showing a message flow between objects in the preferred embodiment of this invention.

Fig.6 is a diagram showing an example of descendant gene creating logic of this invention.

Fig.7 is a diagram showing a message flow between objects in the preferred embodiment of this invention.

Fig.8 is a diagram explaining an example of a gene comparison logic of this invention.

A. Summary

[0020] An embodiment of this invention is now described hereunder with reference to the drawings. Fig.1 shows a distributed network environment 150 executing a mobile object which is generated in accordance

with this invention. Each server 112 - 117 is provided with a place 102 - 107 where a service can be provided to mobile agents 125, 135, 141, etc. The places in the distributed network environment are called a crowd.

[0021] A mobile agent generating part 113 for generating a mobile agent exists in a client system 101. The mobile agent generating part 113 delivers a generated plan to a mobile agent 111 which, in turn, can move through a designated place by executing the plan.

[0022] The mobile agent 125, etc., can contact to another agent (an agent contacting to a mobile agent to provide a service is specifically called an actor agent) existing in each place 102 - 107, send out a request and receive the result of the request. The place supports a contact between agents. Also, the mobile agent 125, etc., can hold the result of the request received from an actor agent as Result 139, 143, 153, 163, continue to move and apply various works on the Result such as composition and split of the Result during movement.

[0023] In the preferred embodiment of this invention, movement, split and extinguishment of a mobile agent are controlled by moderator templates 181 - 187 and a user can simply define a complicated work including split and merge as shown in Fig.2 by combining various templates as required by the mode of processing.

B. Hardware Configuration

[0024] Fig.2 shows a schematic diagram of a hardware configuration of node systems in a preferred embodiment of this invention. The node system 100 comprises a central processing unit (CPU) 1 and a memory 4. The CPU 1 and the memory 4 are connected to a hard disk device 13, 31 as an auxiliary storage via a bus 2, etc. A floppy disk device 20 (or a medium drive device such as an MO, a CD-ROM, etc. 13, 26, 28, 29, 30) is connected to the bus 2 through a floppy disk controller 19 (or a controller such as an IDE controller 25, a SCSI controller 27).

[0025] A floppy disk (or a medium such as an MO, a CD-ROM, etc.) is inserted to the floppy disk device 20 (or a medium drive device such as an MO, a CD-ROM, etc. 26, 28, 29). A computer program code which gives instructions to the CPU and the like in cooperation with an operating system to practice this invention may be recorded in the floppy disk or a recording medium of the hard disk device 13, 30 and a ROM 14, and executed by being loaded in the memory 4. The computer program code may be compressed or divided into pieces for recording in a plurality of media.

[0026] Further, the node system 100 may be a system which is provided with a user interface hardware including a pointing device 7 (a mouse, a joystick and a track ball, etc.) for inputting screen position information, a keyboard 6 for supporting a key input and a display 11, 12 for presenting image data to the user. A speaker 23 receives an audio signal from an audio controller 21 via an amplifier 22 for output as a voice.

[0027] A GUI plan node library (to be described later) which is an input to the node system 100 of this invention is stored in a storage medium such as the floppy disk 24 and the hard disk 30 and inputted to the system via the SCSI interface 27. The GUI plan node library which is an input may be stored in a database of another system. In such case, it is possible to communicate with other computers , etc., to access to a database of another system for obtaining the GUI plan node library via a serial port 15 and a modem or a communication adapter 18 such as a token ring.

[0028] As such, it will be readily understood that this invention may be implemented by a conventional personal computer (PC), a workstation, a computer implemented in home electric appliances such as a television set and a facsimile equipment, and a combination thereof. It should be noted, however, that these components are given for exemplary purpose and it is not meant that all of these components are the indispensable components of this invention. Specifically, because this invention is directed to generation of a mobile agent, those components including the serial port 15, the communication adapter card 18, the audio controller 21, the amplifier 22 and the speaker 23 are not indispensable to one mode of this invention.

[0029] As an operating system, those which support a GUI multi window environment in default, such as Windows (trademark of Microsoft), OS/2 (trademark of IBM), X-WINDOW System (trademark of MIT) on AIX (trademark of IBM) are desirable but the operating system is not limited to any specific operating system environment.

[0030] While Fig.2 shows a system in a stand alone environment, this invention may be implemented in a client/server system in which a client machine is LAN connected to a server machine via Ethernet and a token ring, etc. and the server machine side is provided with a plan split part, etc., to be described later with the rest of functions disposed in the client side. As such, it is a matter of discretion in the design to dispose what function in which of the server machine side and the client machine side. Various modifications of a combination of a plurality of machines, distribution of functions to these machines, etc., and implementation thereof are within the concept of this invention.

C. Processing Components in the System

[0031] Fig.3 is a block diagram showing an embodiment of processing components in the node system 100. The agent 200 holds an ancestor gene 201 and a self gene 203 and obtains a gene 205 from another agent for comparison. The agent 200 is also provided with a child creating module 211, a gene conversion module 213 and a gene comparison module 215. The agent 200 is also provided with an interface to exchange the self gene with other agents.

[0032] The ancestor gene 201 is information to specify

the ancestor (original) of the agent. In the preferred embodiment of this invention, this information is protected from being externally notified or modified itself.

[0033] The self gene 203 is information to specify that the agent itself is how many generations descending from the ancestor. The gene 205 for comparison is information of the self gene 233 of another agent 230 obtained from the agent 230.

[0034] The child creating module 211 is a module for creating a descendant. The child creating module 211 implants a child gene 253 generated by converting the self gene 203 in the created child agent 250. The ancestor gene 201 is also implanted in the child agent 250.

[0035] The gene conversion module 213 converts a gene which is a character string to another gene in response to a request from the child creating module 211 and the gene comparison module. This is a conversion which can be reverse converted. The gene comparison module 215 can identify how many generations the gene is descendant by repeating reverse conversions of that gene.

[0036] The gene comparison module 215 can determine whether or not a gene 205 obtained from another agent is an agent which has a common ancestor and how many generations it is descendant therefrom by obtaining the gene 205 from another agent and repeating reverse conversion of that gene.

D. Object Configuration

[0037] Fig.4 is an object diagram of the mobile agent of this invention. The class of the agent class 300 has data and a method. Each class of a gene converter class 310 and a gene comparison class 320 has a method.

[0038] In the preferred embodiment of this invention, the agent class 300 has data of ancestor gene information 201 (ancestorGene) and self gene information 203 (selfGene). It also has methods "get ancestorGene" and "get selfGene" to obtain ancestor gene information 201 and self gene information 203.

[0039] A "create child" method is a method for creating a child agent while a "create clone" method is a method for creating a clone of itself. A "create clone" method has an ancestor gene (ancestorGene) and a self gene of a child agent (childGene) as an argument.

[0040] The gene converter class 310 has a convertGene method for converting gene information in accordance with the generation. The convertGene method has converted gene information (gene) and generation information (n) as arguments. The gene comparison class 320 has a compareGene method for comparing gene information. The compareGene method has gene information (gene) as an argument.

E. Creation of Child Agent

[0041] Fig.5 is a diagram showing a major message flow of each class upon creating a child agent in the preferred embodiment of this invention.

[0042] When the createChild method is called upon for creating a child agent, a self gene is first obtained from the agent. Next, the gene comparison class 320 is inquired of how many generations the agent is descendant.

[0043] The ancestor gene 201 is then obtained and the gene converter class 310 is asked to generate the gene 253 of the child agent 250. The argument then is generation information of its generation + 1 and ancestor gene information. When the gene 253 of the child agent 250 is delivered from the gene converter class 310, a clone is created (create a copy of the agent own) using the gene 253 and the ancestor gene information as an argument. The child agent 250 is thus created.

[0044] Fig.6 is a diagram showing an example of self gene creating logic (of descendant) of this invention. The example is explained by 16 bits string to simplify the explanation.

[0045] The leftmost 8 bits are an ID which is common to an agent family. In the preferred embodiment of this invention, this ID is assigned a value which is unique on the global network. For example, the ID may be created by composing an internet URL, a port number and a serial number (assigned each time when creating an original mobile agent).

[0046] The rightmost 8 bits are a bit string which specify the generation. In this example, the original agent (originator agent) is given gene information "10101101" (coincides the ancestor gene). The child (the second generation) agent is give gene information "01010010" which is an inversion of the rightmost 8 bits.

[0047] Further, its child (grand child: the third generation) agent is given gene information "01011101" by inverting the rightmost 4 bits. The fourth generation agent is given "01011110" by inverting the rightmost 2 bits and the fifth generation agent is given "01011111" by inverting the rightmost 1 bit.

[0048] In this example, up to the fifth generation can be managed. However, it is enough for the gene information to be information which allows determination of the level of generation from the ancestor and whether or not the descendant of that ancestor. The gene information may be information which encrypts the ancestor gene information and the generation information (or only the ancestor gene information) by a predetermined algorithm.

F. Comparison of Gene

[0049] Fig.7 is a diagram showing a major message flow of classes upon comparing genes in the preferred embodiment of this invention.

[0050] The agent 200 requests another agent on a same place to deliver its self gene (get selfGene). The gene thus obtained is stored as a gene 205 to be compared. In the preferred embodiment of this invention, the

agent family ID is first identified. The gene 205 to be compared is then delivered to the gene comparison class 320 to determine the generation.

[0051] The gene comparison module 215 reverse converts the gene to compare it with the ancestor gene. If the compare is equal in the Nth comparison, it is identified as a Nth generation descendant from the ancestor. In the preferred embodiment of this invention, when it is identified as an ancestor, the gene comparison class 320 returns N while it returns error information such as a low value otherwise (when the comparison fails to match the ancestor in a predetermined number of conversions, i.e., when it is determined that the agent is not a descendant).

[0052] Fig.8 is a diagram explaining an example of a reverse conversion logic of gene in this invention. The gene of the fifth generation described in Fig.5 is used. In this conversion also, the gene is inverted in the sequence of the rightmost 8 bits, 4 bits, 2 bits and 1 bit in the manner similar to Fig.6 and compared to the ancestor gene 201 each time it is converted. As shown in the figure, it is determined that the gene matches the ancestor gene 201 in the fourth conversion. In other words, it can be identified that the mobile agent has a common ancestor and how many generations it is descendant from the ancestor.

[0053] Similarly, the second generation matches the ancestor gene 201 in the first conversion ("01010010" => "10101101"), the third generation matches in the second conversion ("01011101" => "10100010" => "10101101"), and the fourth generation matches in the third conversion ("01011110" => "10100001" => "10101110" => "10101101").

[0054] As described in the above, this invention allows a mobile agent to identify its ancestor, children, grand children and sibling, etc., and to flexibly cooperate with them in working by implanting gene information in a mobile agent while maintaining a high security.

Claims

1. A mobile agent which is adapted to interoperate with other mobile agents at the same network location, comprising:

   (a) self gene information from which ancestor gene information and generation information can be derived by a predetermined conversion logic,

   (b) a gene conversion module for generating child agent gene information from which said ancestor gene information and said generation information can be derived by a predetermined conversion logic, said child agent gene information being implanted in a child agent when creating a child agent of said mobile agent, and

   (c) a gene comparison module for determining whether or not the result of a conversion of gene information obtained from another agent corresponds to said ancestor gene information.

2. An object held by a mobile agent which is adapted to interoperate with other mobile agents at the same network location, comprising:

   (a) self gene information from which ancestor gene information and generation information can be derived by a predetermined conversion logic,

   (b) a gene conversion module for generating child agent gene information from which said ancestor gene information and said generation information can be derived by a predetermined conversion logic, said child agent gene information being implanted in a child agent when creating a child agent of said mobile agent, and

   (c) a gene comparison module for determining whether or not the result of a conversion of gene information obtained from another agent corresponds to said ancestor gene information.

3. A method of creating a child agent based on a mobile agent existing at a computer network location, the method comprising;

   (a) a step of deriving generation information of said mobile agent based on self gene information and ancestor gene information,

   (b) a step of generating gene information of a child agent based on said derived generation information and said ancestor gene information, and

   (c) a step of creating a child agent which contains the generated gene information of the child agent and said ancestor gene information.

4. A method for allowing a first mobile agent existing at a computer network location to identify whether a second mobile agent existing at the same network location has the same ancestor gene information as is held in said first mobile agent, the method comprising;

   (a) a step of obtaining gene information from said second mobile agent,

   (b) a step of converting said obtained gene information by a predetermined logic, and

(c) a step of determining whether or not the result of the conversion corresponds to said ancestor gene information held in said first mobile agent.

5. A mobile agent which is adapted to interoperate with other mobile agents at the same computer network location, and from which child agents can be generated, comprising;

(a) first gene conversion logic for converting self gene information held by said mobile agent to derive ancestor gene information and generation information,

(b) second gene conversion logic for generating child agent gene information from which said ancestor gene information and said generation information can be derived by said first gene conversion logic, said child agent gene information being implanted in a child agent when creating a child agent of said mobile agent,

(c) means for interrogating another mobile agent at the same computer network location to obtain its gene information, and

(d) a gene comparison module for determining whether or not the result of a conversion of gene information obtained from another agent corresponds to said ancestor gene information.

6. A mobile agent according to claim 5, wherein the self gene information is held by the mobile agent in an encrypted form and the converting of said self gene information and child agent gene information includes decrypting the encrypted self gene information.

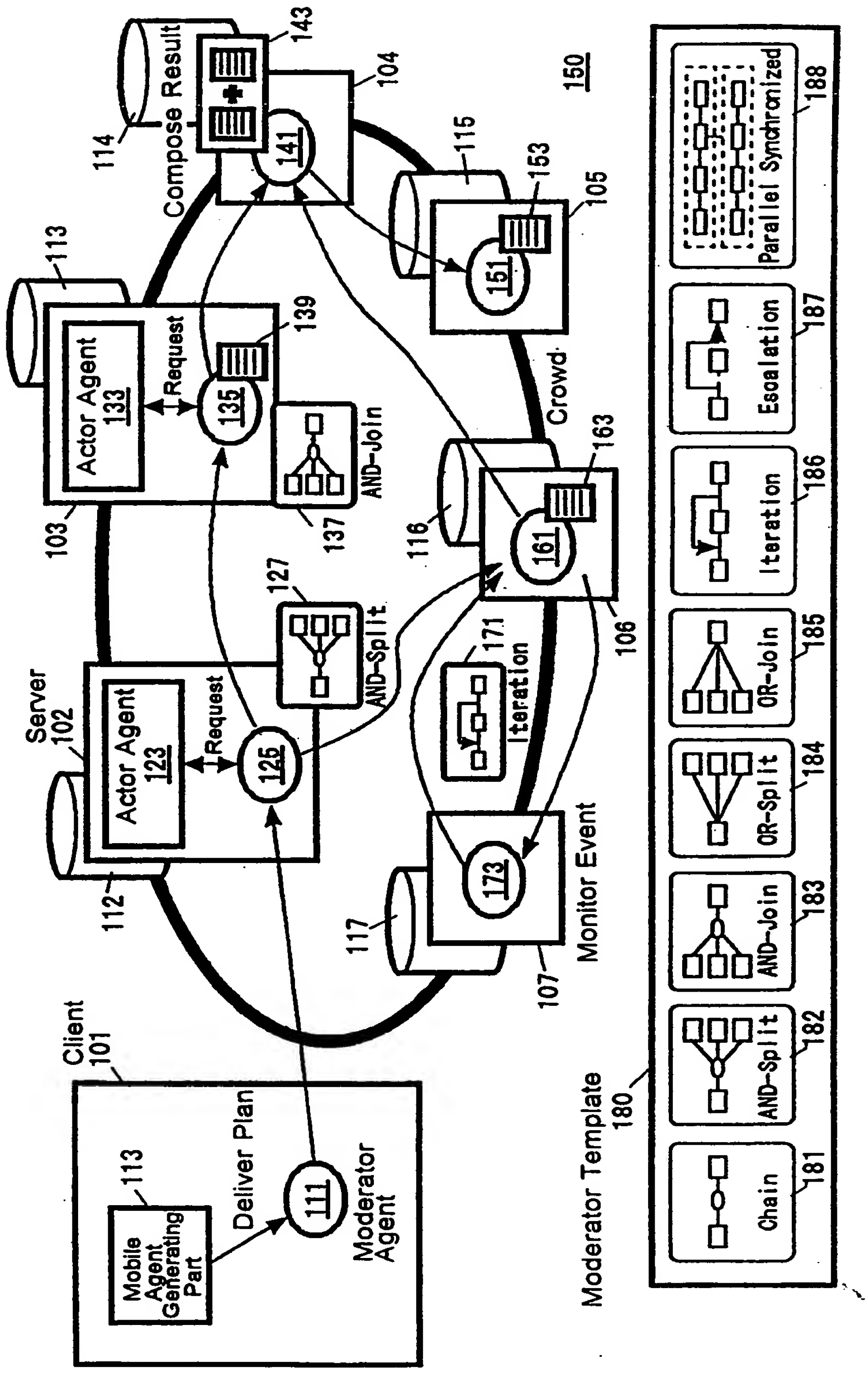
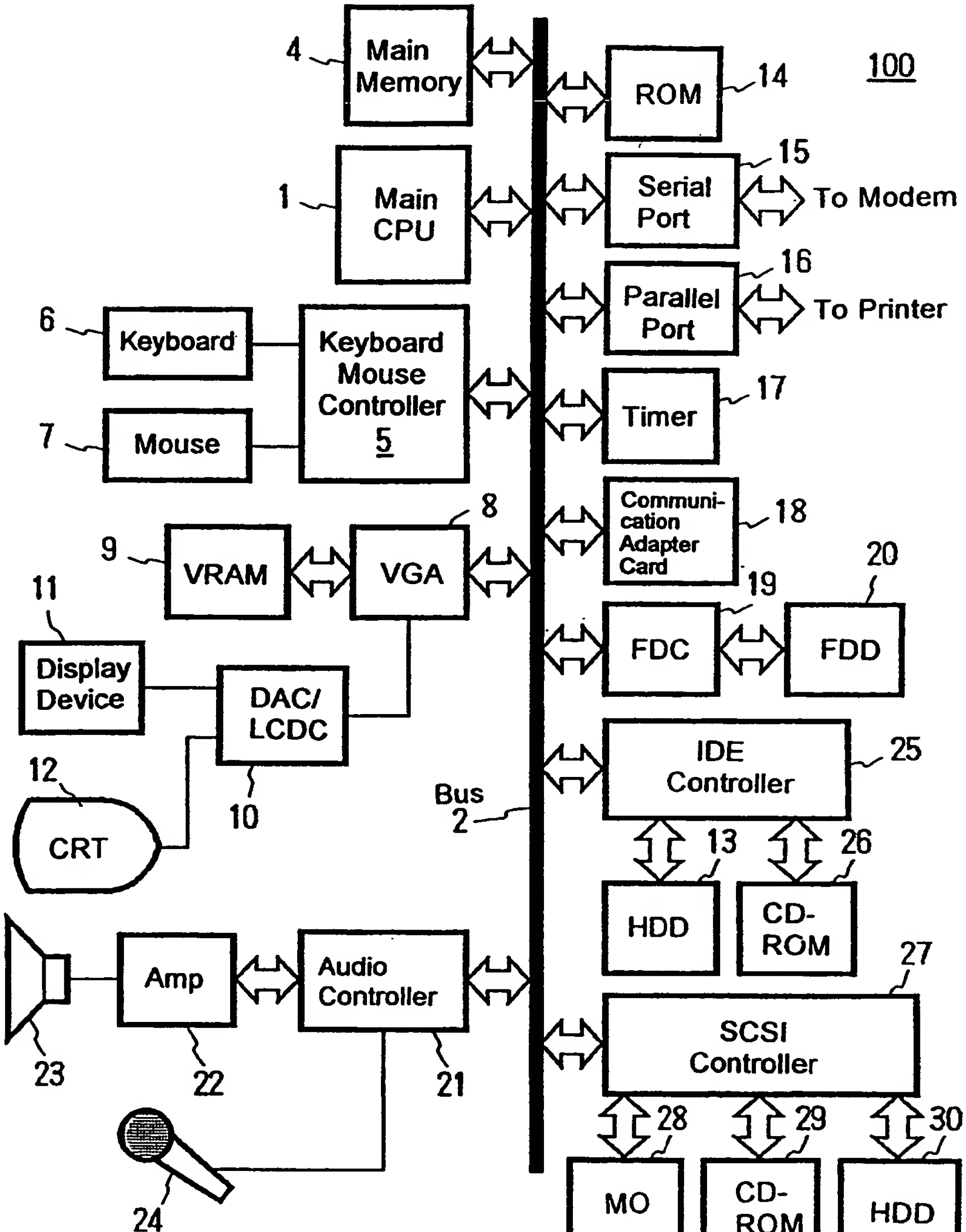

Client
101
Mobile Agent Generating Part
113
Deliver Plan
111
Moderator Agent
Server
102
Actor Agent
123
Request
125
112
AND-Split
127
Actor Agent
133
Request
135
139
103
AND-Join
137
113
Compose Result
114
141
143
104
115
151
153
105
150
Crowd
163
161
116
106
Iteration
171
173
117
107
Monitor Event
Moderator Template
180
Chain
181
AND-Split
182
AND-Join
183
OR-Split
184
OR-Join
185
Iteration
186
Escalation
187
Parallel Synchronized
188

FIG. 1



Main Memory
4
ROM
14
100
Main CPU
1
Serial Port
15
To Modem
Parallel Port
16
To Printer
6
Keyboard
Keyboard Mouse Controller 5
7
Mouse
Timer
17
8
9
VRAM
VGA
Communi-cation Adapter Card
18
11
19
20
Display Device
DAC/ LCDC
FDC
FDD
12
10
IDE Controller
25
CRT
Bus 2
13
26
HDD
CD-ROM
27
Amp
Audio Controller
SCSI Controller
23
22
21
28
29
30
MO
CD-ROM
HDD
24

FIG. 2

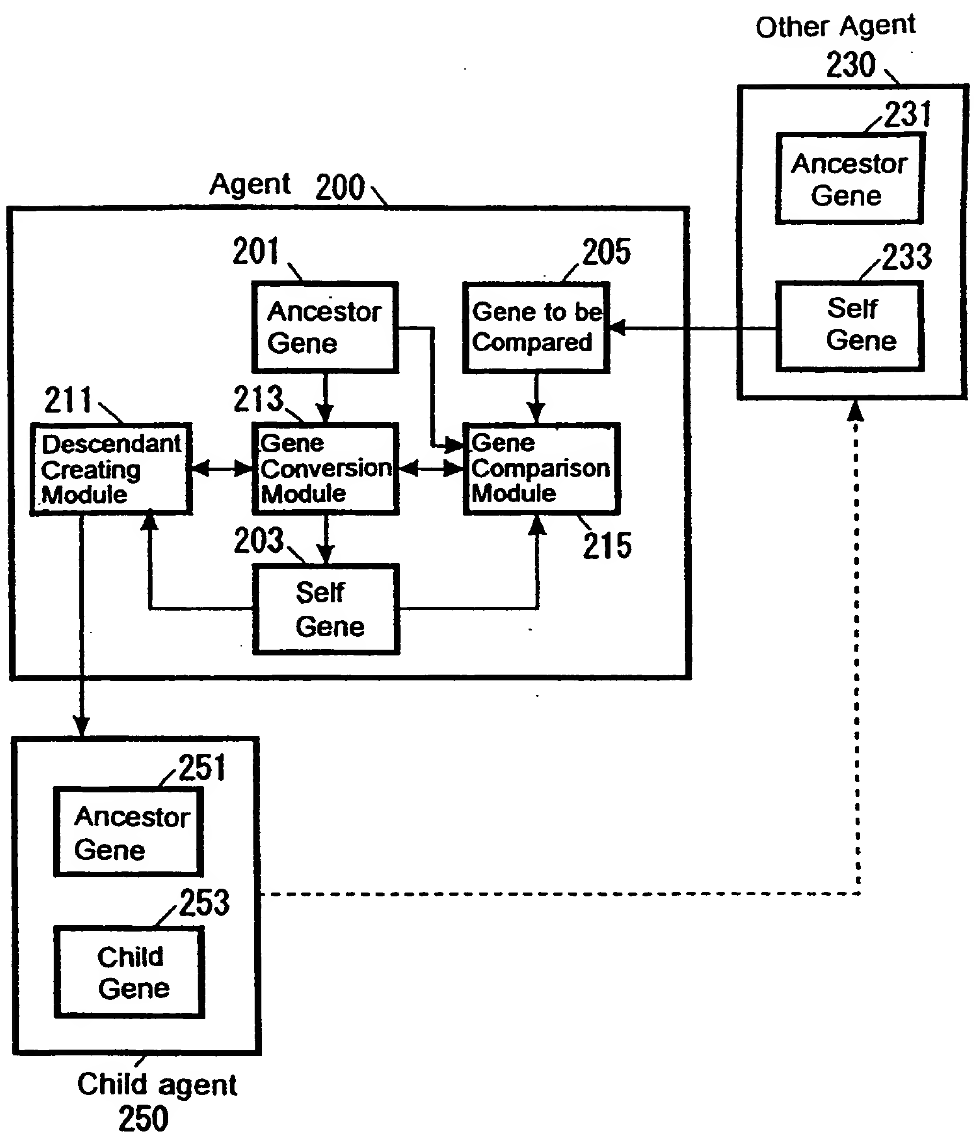

Other Agent
230
231
Ancestor Gene
233
Self Gene
Agent 200
201
Ancestor Gene
205
Gene to be Compared
211
Descendant Creating Module
213
Gene Conversion Module
Gene Comparison Module
215
203
Self Gene
251
Ancestor Gene
253
Child Gene
Child agent
250

FIG. 3

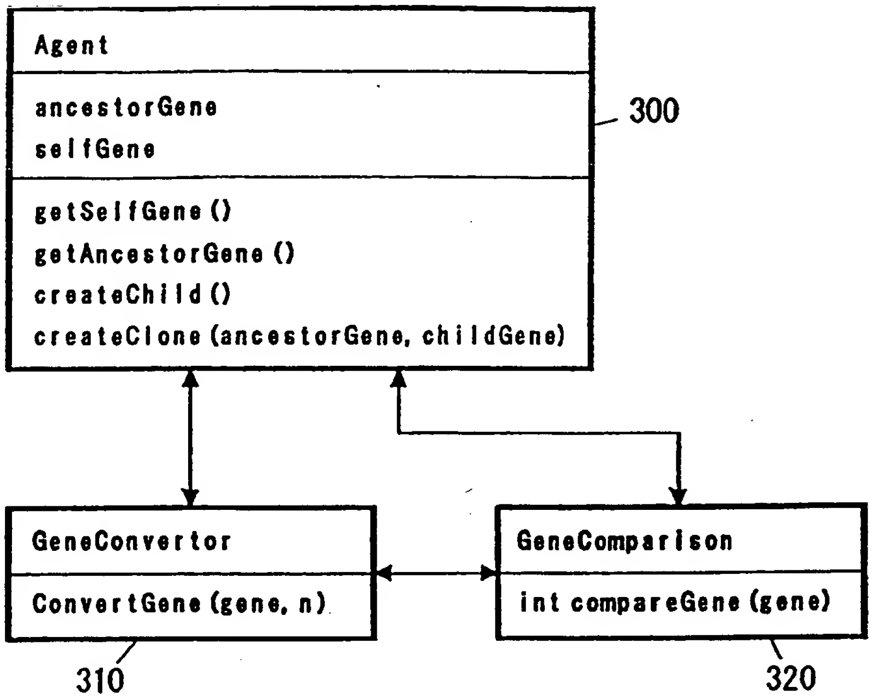

Agent
ancestorGene
selfGene
getSelfGene ()
getAncestorGene ()
createChild ()
createClone (ancestorGene, childGene)
300
GeneConvertor
ConvertGene (gene, n)
GeneComparison
int compareGene (gene)
310
320

FIG. 4

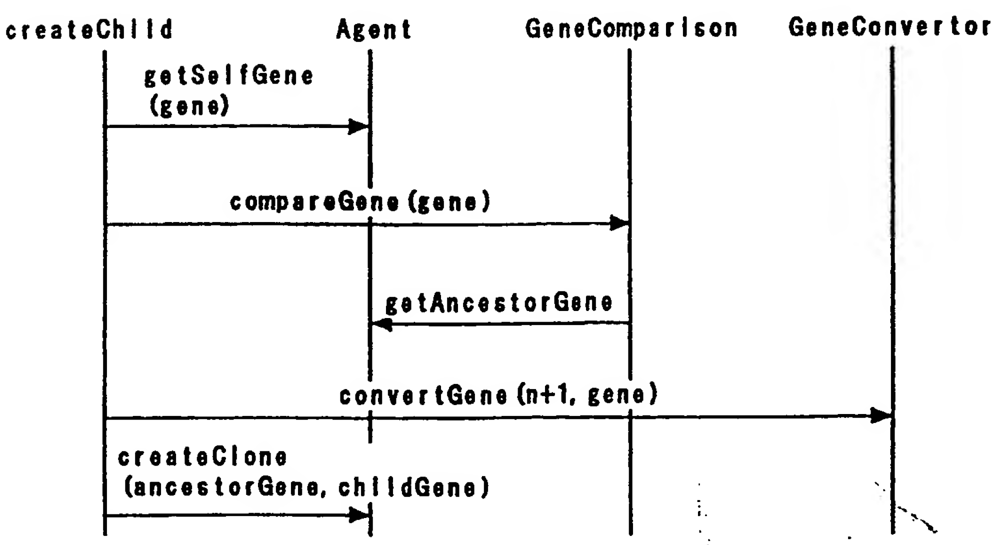

createChild
Agent
GeneComparison
GeneConvertor
getSelfGene
(gene)
compareGene (gene)
getAncestorGene
convertGene (n+1, gene)
createClone
(ancestorGene, childGene)

FIG. 5

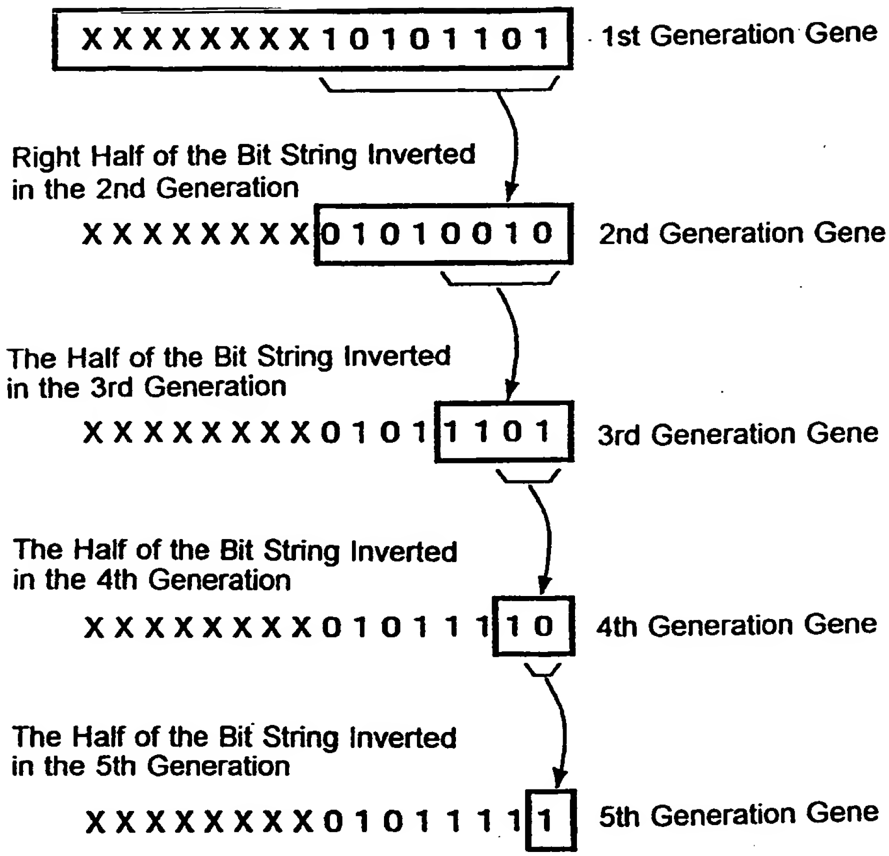

XXXXXXXX10101101
1st Generation Gene
Right Half of the Bit String Inverted
in the 2nd Generation
XXXXXXXX01010010
2nd Generation Gene
The Half of the Bit String Inverted
in the 3rd Generation
XXXXXXXX01011101
3rd Generation Gene
The Half of the Bit String Inverted
in the 4th Generation
XXXXXXXX01011110
4th Generation Gene
The Half of the Bit String Inverted
in the 5th Generation
XXXXXXXX01011111
5th Generation Gene

FIG. 6

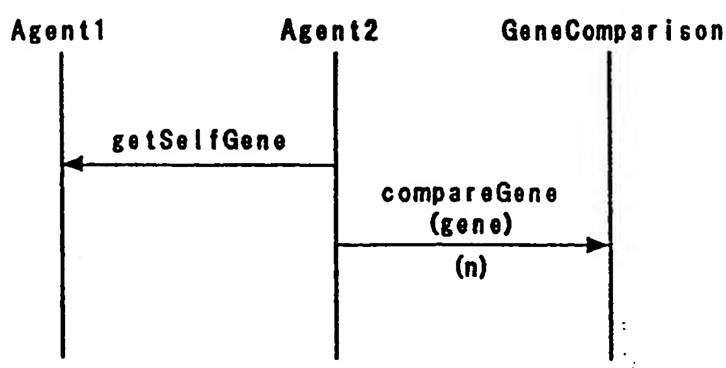

Agent1
Agent2
GeneComparison
getSelfGene
compareGene
(gene)
(n)

FIG. 7

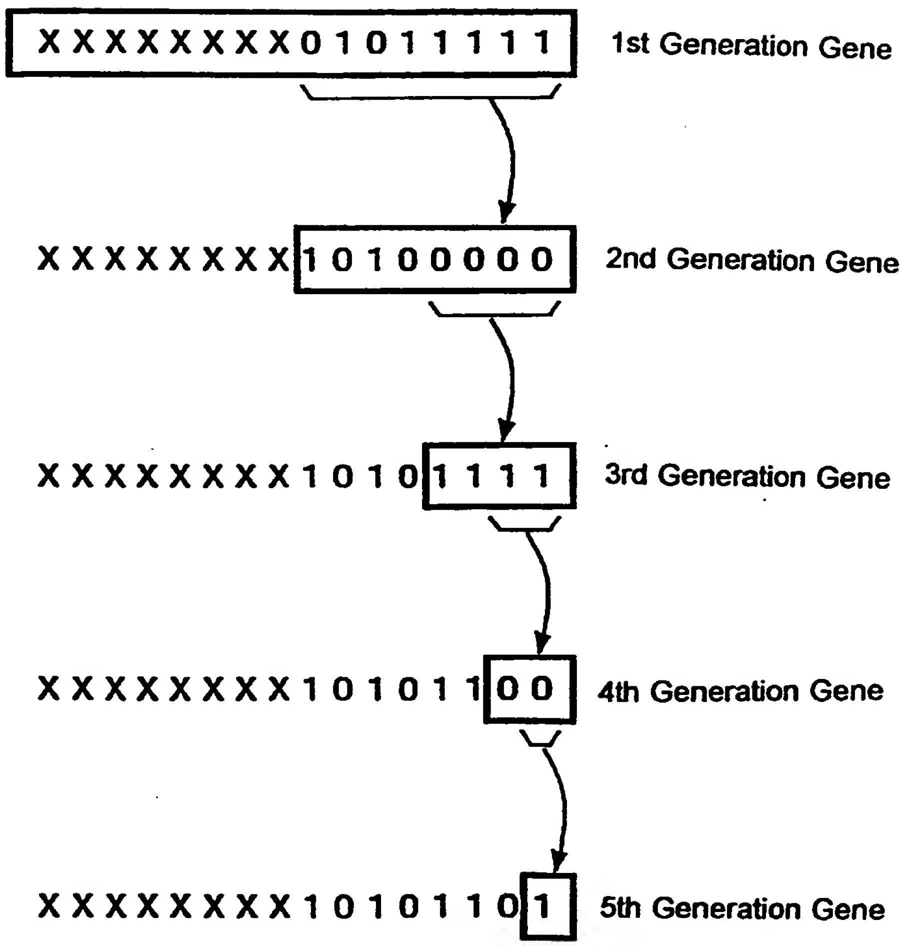

XXXXXXXX01011111
1st Generation Gene
XXXXXXXX10100000
2nd Generation Gene
XXXXXXXX10101111
3rd Generation Gene
XXXXXXXX10101100
4th Generation Gene
XXXXXXXX10101101
5th Generation Gene

FIG. 8

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 30 1899

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	H. PEINE: "An introduction to mobile agent programming and the Ara system" ZRI-REPORT, 'Online! January 1997 (1997-01), pages 1-67, XP002109013 kaiserslautern, Germany Retrieved from the Internet: <URL:http://www.uni-kl.de/AG-Nehmer/Projekte/Ara/documents.html> 'retrieved on 1999-07-13! * page 18-22 * -----	1-6	G06F9/46
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)
			G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	13 July 1999	Michel, T

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)